

INVESTIGACIÓN LA FORMULA DE LA DISTANCIA EUCLIDIANA?

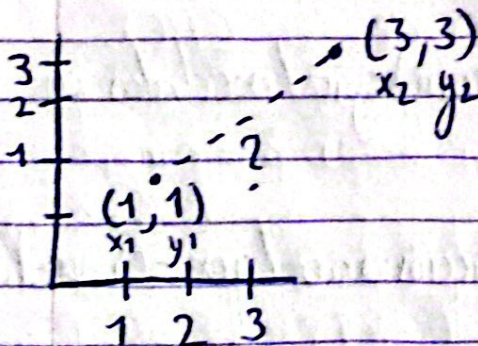
$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

p, q = dos puntos en el espacio n euclidiano

q_i, p_i = Vectores euclidianos, a partir del origen del espacio (punto inicial)

n = espacio n

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



$$d = \sqrt{(3-1)^2 + (3-1)^2}$$

Realice el diagrama de flujo y pseudocódigo.

Diagrama De Flujo: Representa la esquematización grafica de un algoritmo, el cual muestra graficamente los pasos o procesos a seguir para alcanzar la solución de un problema.

Pseudocódigo: Descripción detallada y legible de lo que debe hacer un programa.

2: Algoritmo para calcular la distancia entre los puntos $(1,1)$ $(3,3)$

1*

Autor: Diana Aula

Fecha: 28/01/2025

Pseudocódigo y algoritmo del cálculo de la distancia
Euclidiana*

1. Comienzo

2. Solicitar las coordenadas del primer punto

2.1 Solicitar y guardar x_1

2.2 Solicitar y guardar y_1

3. Solicitar las coordenadas del ~~segundo~~ segundo punto

3.1 Solicitar y guardar x_2

3.2 Solicitar y guardar y_2

4. Guardar en variable aux1 la resta de $x_2 - x_1$

5. Guardar en variable aux2 la resta de $x_2 - y_2$

6. Guardar en variable aux3 el cuadrado de aux1

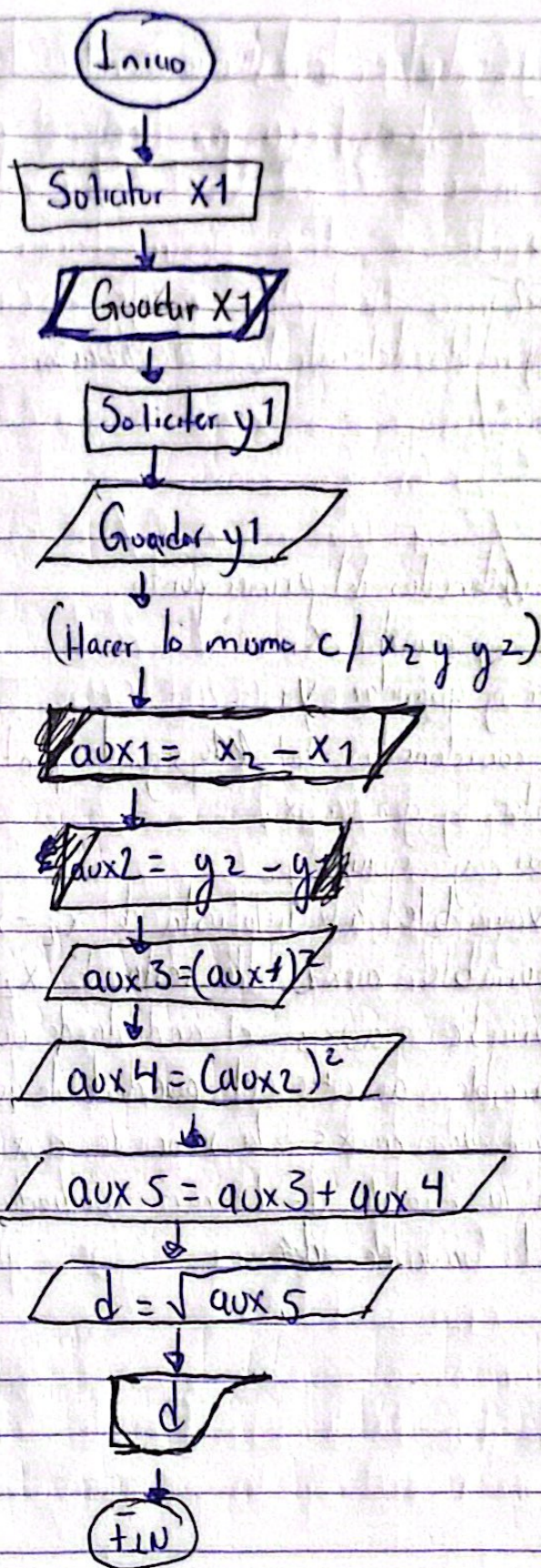
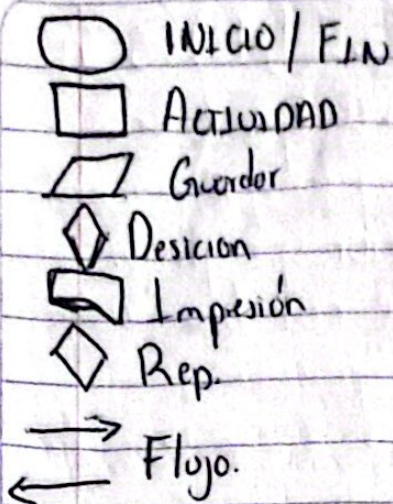
7. Guardar en variable aux4 el cuadrado de aux2

8. Guardar en variable aux5 la suma de aux3 y aux4

9. Guardar en la variable d = La raíz cuadrada de aux5

10. Imprimir la variable d

11. Fin.



¿QUE ES UNA VARIABLE EN INFORMÁTICA Y PROGRAMACIÓN?

* **VARIABLE EN INFORMÁTICA:** En informática una variable es un espacio de memoria reservado que se utiliza para almacenar y presentar un valor. Puede contener números, texto, listas u otros tipos de datos.

* **VARIABLE EN PROGRAMACIÓN:** Una variable representa un contenedor o un espacio en la memoria física o virtual en una computadora, donde se almacenan distintos tipos de datos (valores) durante la ejecución de un programa. Se caracteriza por el **ALMACENAMIENTO DE DATOS EN LENGUAJE C: NOMBRE, TIPO DE DATOS QUE ALMACENA, CUANTO ESPACIO OCUPA EN LA MEMORIA.**

* **PRIMITIVOS:** Se llaman primitivos porque están definidos por el compilador

- Tipos ENTEROS: Char, short, int, long, long long y enum
- Tipos REALES: float, double, y long double.

* Cada tipo de dato entero tiene un rango definido, lo que significa que tiene un número de bits (unos y ceros), lo que significa que de manera subyacente un número entero como por ejemplo 7 puede ser representado como 0000000000 000111.

1- **Int:** Utilizado para almacenar números enteros, su tamaño suele ser de 4 bytes, aunque puede variar según la implementación del compilador.

2- **Float:** Utilizado para almacenar números de punto flotante de precisión simple, generalmente ocupa 4 bytes.

3- **Double:** Utilizado para almacenar números de punto flotante de doble precisión, suele ocupar 8 bytes.

4- Char: Utilizado para almacenar un solo caracter. Ocupa 1 byte.

~~5- Void: No es un tipo de dato que almacene valores, pero se utiliza para indicar que una función no retorna ningún valor.~~

6- Bool: Introducido en el estándar C99, se utiliza para almacenar valores booleanos (0 para falso y 1 para verdadero). Ocupa 1 byte.

* DERIVADOS... Los tipos de datos derivados son aquellos que se construyen a partir de los datos básicos.

1- Arrays (Arreglos):

* Un array es una colección de elementos del mismo tipo almacenados en ubicaciones de memoria antiguas.

~~2- Structs (Estructuras):~~

Es un conjunto de variables de diferentes tipos agrupadas bajo un mismo nombre.

3- Unions (Uniones): Una union es similar a una estructura, pero todas las variables, comparten la misma ubicación de memoria.

4- Enumeration (Enumeraciones):

Una enumeración es un tipo de dato que consiste en un conjunto de constantes enteras.

5- Pointers (Apuntadores):

1 deema

M

A

BACK 2
SCHOOL

Un apuntador es una variable que almacena la direccion de memoria de otra variable.

QUE ES UNA CONSTANTE?

Una constante en programacion es un valor fijo que un programa no puede modificar mientras se esta ejecutando.

H=...

```
#include <stdio.h>
```

```
int main ( )
```

```
{ //llave
```

```
// Zona de declaracion de variables
```

```
int x1, x2, y1, y2;
```

```
printf("DAME EL VALOR DE x1");
```

```
scanf("%d", &x1);
```

```
printf("DAME EL VALOR DE x2");
```

```
scanf("%d", &x2);
```

```
printf("DAME EL VALOR DE y1");
```

```
scanf("%d", &y1);
```

```
printf("DAME EL VALOR DE y2");
```

```
scanf("%d", &y2);
```

```
return 0;
```

```
} //llave
```